



UNIVERSITÀ DEGLI STUDI DI PALERMO
Corso di ingegneria informatica e delle telecomunicazioni, dipartimento DIID

Software Design Document

Di Andrea Vaiuso

Ingegneria informatica e delle
Telecomunicazioni

Sommario

1. Generali.....	2
1.1 Dizionario.....	2
2. Sistema Proposto.....	2
2.1 Panoramica.....	2
2.2 Requisiti Funzionali.....	2
2.3 Requisiti non funzionali.....	3
3. Modelli di sistema	3
3.1 Attori coinvolti.....	3
3.2 Diagrammi dei casi d'uso.....	4
3.3 Diagrammi delle classi.....	12
3.4 Diagrammi di sequenza	20
4. Object Design.....	27
4.1 Trade off.....	27
4.2 Descrizione dei model.....	27
4.3 Descrizione dei view	27
4.4 Descrizione dei control	28
5. Package.....	29
5.1 Organizzazione dei package.....	29
6. System Design	30
6.1 Hardware – Software mapping.....	30
6.2 Decomposizione del sistema.....	30
7. Database Design	31
7.1 Diagramma ER ristrutturato	31

1. Generali

1.1 Dizionario

EasyPay: Servizio di pagamento online con il quale è possibile inviare e ricevere pagamenti tramite account registrati e non. Vedasi la definizione completa nella panoramica del prodotto, cap 2 par 1

Server: Applicazione remota che si occupa di gestire le comunicazioni col DBMS ed i servizi di notifica via email e Whatsapp

Utente: Attore principale che fa uso dei vari servizi offerti dall'applicazione tramite un account che è possibile creare all'interno dell'applicazione

2. Sistema Proposto

2.1 Panoramica

EasyPay è un sistema per effettuare pagamenti e scambi di denaro tra utenti registrati. Un utente registrato ha il proprio portafoglio e può registrare carte di credito, debito e conti correnti bancari per effettuare pagamenti. Possono essere effettuati pagamenti ad altri utenti (identificati dall'indirizzo email) o a enti commerciali (identificati dalla ragione sociale). Possono essere inviate richieste di pagamento ad altri utenti e possono essere creati pagamenti ricorrenti per gli enti commerciali. Inoltre è possibile inviare pagamenti e richieste di pagamento anche ad utenti non registrati che attraverso dei codici univoci potranno riscattare o pagare le somme dopo essersi registrati. EasyPay fornisce anche un sistema di notifiche per avvisare gli utenti delle varie operazioni.

2.2 Requisiti Funzionali

Le funzionalità sono divise in due diversi settori:

- **Funzionalità di gestione degli account:**
 - Accesso utente registrato: Gli utenti attraverso un sistema di login potranno accedere ai vari servizi
 - Registrazione nuovo utente: Nuovi utenti possono essere registrati per usufruire dei servizi
 - Recupero password utente: Gli utenti che hanno dimenticato la password potranno recuperarla richiedendola via email
 - Preferenze dati utente: Gli utenti possono modificare i propri dati anagrafici e le impostazioni sulla ricezione delle notifiche via email o whatsapp
 - Gestione carte e banche: Gli utenti potranno collegare e scollegare carte e conti bancari pre esistenti ed utilizzarli per effettuare le operazioni di pagamento ed accredito
- **Funzionalità di gestione dei pagamenti:**

- Invio pagamento a utenti o enti commerciali: Gli utenti possono inviare pagamenti ad utenti registrati e non (tramite un codice per riscattare la cifra), oppure ad enti commerciali utilizzando il proprio portafoglio EasyPay oppure tramite carta o conto corrente bancario registrati.
- Richiesta pagamento ad utenti: Gli utenti possono richiedere pagamenti ad altri utenti registrati e non.
- Creazione pagamenti periodici ad enti commerciali: Gli utenti possono inviare somme prestabilite di denaro ad enti commerciali con cadenza settimanale, mensile, oppure annuale.

2.3 Requisiti non funzionali

La programmazione del software sarà guidata dai seguenti requisiti non funzionali:

- **Consistenza:**
Il software garantirà la consistenza dei dati tramite la gestione di essi attraverso l'utilizzo di MySQL
- **Implementazione:**
Il software verrà implementato tramite l'utilizzo del linguaggio di scripting Javascript sia per il lato server che per il lato client, con l'utilizzo di HTML e CSS per la visualizzazione dell'interfaccia attraverso il browser
- **Intuitività:**
Le funzionalità del software saranno chiare e di facile utilizzo per gli utenti grazie ad un'interfaccia minimale e semplice
- **Rispetto delle normative legali:**
I dati sensibili saranno memorizzati in maniera sicura e le password criptate
- **Sicurezza:**
Tramite un sistema di login gli utenti potranno accedere ai propri dati in sicurezza

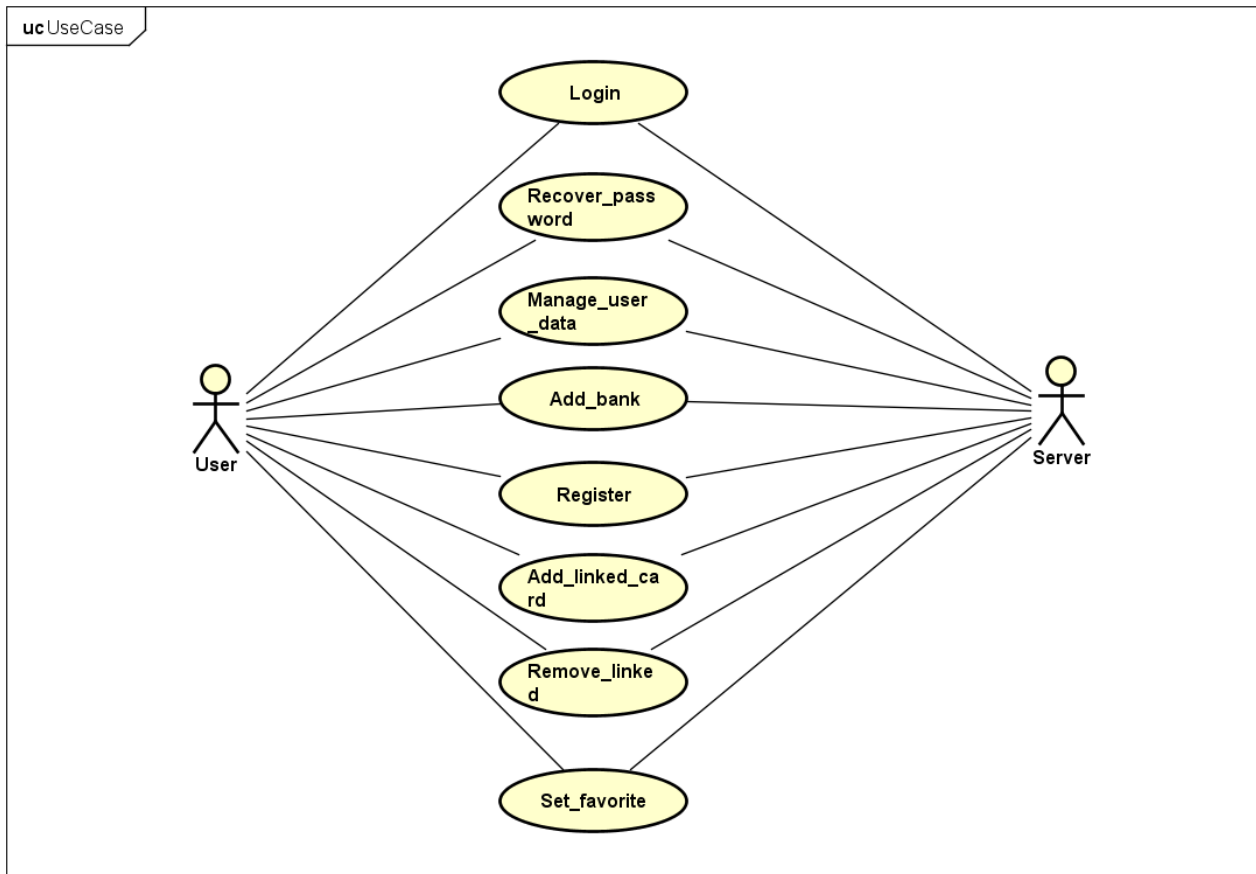
3. Modelli di sistema

3.1 Attori coinvolti

- **Utente:**
Attore che viene coinvolto nelle funzionalità che il sistema offre ai clienti
- **Server:**
Attore che gestisce la comunicazione col DBMS ed il servizio di messaggistica

3.2 Diagrammi dei casi d'uso

Diagrammi dei casi d'uso per la gestione degli account



Nome caso d'uso:	Login
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la finestra di login
Flusso degli eventi:	<ol style="list-style-type: none">1. L'utente immette la sua email con cui si è registrato e la password e clicca su "login"2. I dati vengono inviati al server che si connette al DBMS per verificarne l'attendibilità. Nel caso di mancata corrispondenza viene mostrato un messaggio di errore
Post condizioni:	Viene conferito all'utente l'accesso all'area personale

Nome caso d'uso:	Register
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la finestra di login
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "Registrati" e viene mostrata una pagina per registrarsi 2. L'utente inserisce l'email, la password, il nome, il cognome, l'indirizzo e il numero di telefono e fa click su conferma. Nel caso in cui uno o più campi non fossero validi viene mostrato un messaggio di errore 3. Il software invia al server le informazioni e le inserisce nel database per permettere l'accesso in futuro
Post condizioni:	Viene mostrato un messaggio di conferma e viene nuovamente caricata la pagina di login

Nome caso d'uso:	Recover_password
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la finestra di login
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "recupera password" 2. Il software invia una richiesta al server che verifica col database l'esistenza dell'indirizzo email. Se l'indirizzo email esiste viene inviata al suddetto indirizzo una email con una nuova password.
Post condizioni:	Il sistema mostra un messaggio di conferma

Nome caso d'uso:	Manage_user_data
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "modifica dati" 2. L'utente modifica le informazioni desiderate e fa click su conferma. L'utente può anche disattivare o attivare le notifiche via email, Whatsapp o SMS 3. Il software invia al server le informazioni e ne verifica la validità. In caso siano valide vengono aggiornati i dati nel DBMS, altrimenti viene chiesto all'utente di cambiare i dati non validi
Post condizioni:	Viene mostrato un messaggio di conferma

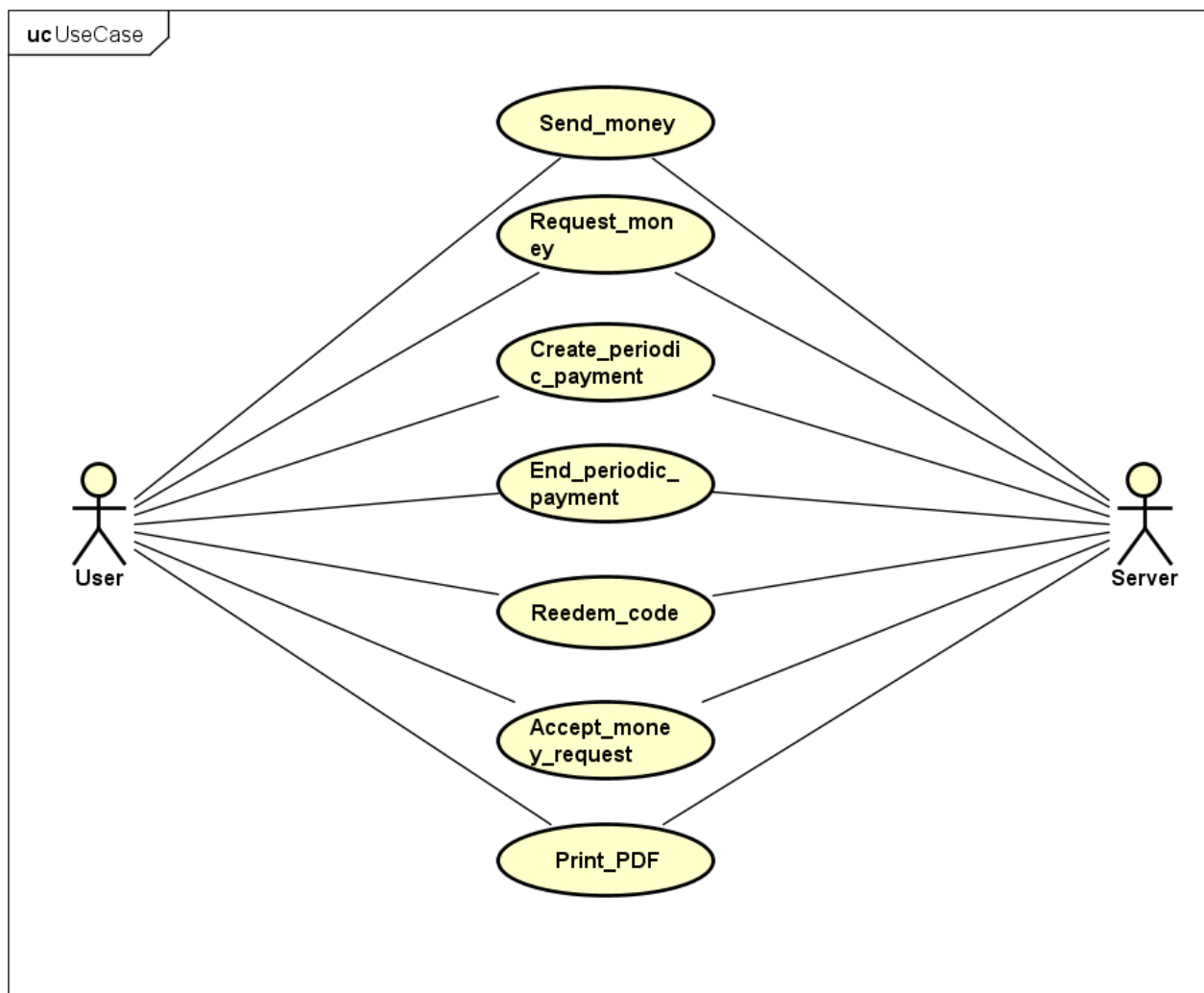
Nome caso d'uso:	Add_linked_card
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "Gestisci carte" 2. Il sistema mostra la pagina di gestione carte 3. L'utente tramite gli appositi form inserisce i dati sulla carta desiderata e fa click su "aggiungi carta" 4. Il sistema si collega al dbms della banca (in simulazione) per la verifica della correttezza dei dati ed inserisce nel database le informazioni
Post condizioni:	Viene mostrato un messaggio di conferma e viene ricaricata la pagina

Nome caso d'uso:	Add_bank
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "Gestisci carte" 2. Il sistema mostra la pagina di gestione carte 3. L'utente tramite gli appositi form inserisce i dati sul conto corrente desiderato e fa click su "aggiungi banca" 4. Il sistema si collega al dbms della banca (in simulazione) per la verifica della correttezza dei dati ed inserisce nel database le informazioni
Post condizioni:	Viene mostrato un messaggio di conferma e viene ricaricata la pagina

Nome caso d'uso:	Remove_linked
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "Gestisci carte" 2. Il sistema mostra la pagina di gestione carte 3. L'utente sceglie la carta o il conto corrente da rimuovere e clicca il tasto "scollega" 4. Il sistema comunica al server le informazioni e viene cancellato il collegamento dal database
Post condizioni:	Viene mostrato un messaggio di conferma e viene ricaricata la pagina

Nome caso d'uso:	Set_favorite
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "Gestisci carte" 2. Il sistema mostra la pagina di gestione carte 3. L'utente clicca la carta o il conto corrente da usare come preferito 4. Il sistema comunica il cambiamento al server che si collega al dbms per aggiornare il database
Post condizioni:	Viene ricaricata la pagina

Diagrammi dei casi d'uso per la gestione dei pagamenti



Nome caso d'uso:	Send_money
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "Invia Denaro" 2. Il sistema mostra la schermata per inviare denaro 3. L'utente sceglie il metodo di pagamento, sceglie se inviare il pagamento ad una persona oppure un ente commerciale ed inserisce rispettivamente l'email o la ragione sociale corrispondente, inserisce la causale e l'importo e fa click su "invia pagamento" 4. Vengono inviati i dati al server che verifica col database l'esistenza della email nel caso in cui fosse stata selezionata la persona, o la ragione sociale nel caso di un ente commerciale. Nel primo caso, se l'email esiste viene vengono aggiornati i portafogli

	dei rispettivi account. Se non esiste viene generato un codice che viene inserito nella sezione riscatta codice, e viene inviato via email al contatto che non esiste, assieme ad un link per registrarsi. Inoltre viene inviata una notifica all'utente che riceve il pagamento per aggiornarlo sulla modifica del saldo
Post condizioni:	Viene mostrato un messaggio di conferma e viene ricaricata la pagina iniziale

Nome caso d'uso: Request_money	
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "Richiedi Denaro" 2. Il sistema mostra la schermata per la richiesta di denaro 3. L'utente inserisce l'indirizzo di destinazione, la causale e fa click su "Richiedi denaro" 4. Il sistema invia i dati al server e controlla l'esistenza dell'indirizzo di destinazione e ne mostra le informazioni. Nel caso in cui l'utente non esista mostrerà l'indirizzo email 5. L'utente conferma l'operazione 6. Il sistema invia una mail di notifica all'utente a cui viene effettuata la richiesta, con un link che permette di effettuare il pagamento richiesto.
Post condizioni:	Viene mostrato un messaggio di conferma e viene ricaricata la pagina iniziale

Nome caso d'uso: Create_periodic_payment	
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "Pagamento periodico" 2. Il sistema mostra la schermata per la creazione di un pagamento periodico 3. L'utente inserisce il metodo di pagamento, la ragione sociale dell'ente commerciale, la periodicità, il numero di pagamenti limit, la data di inizio e l'importo e clicca "Avvia pagamento periodico" 4. Il sistema verifica col database l'esistenza della ragione sociale ed inserisce i dati riguardanti il nuovo pagamento periodico nel database

Post condizioni:	Viene mostrato un messaggio di conferma e viene ricaricata la pagina iniziale
-------------------------	---

Nome caso d'uso:	End_periodic_payment
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "Fine pagamento periodico" al pagamento periodico desiderato 2. Il sistema comunica al database di eliminare le informazioni rispetto quel pagamento periodico
Post condizioni:	Viene mostrato un messaggio di conferma e viene ricaricata la pagina iniziale

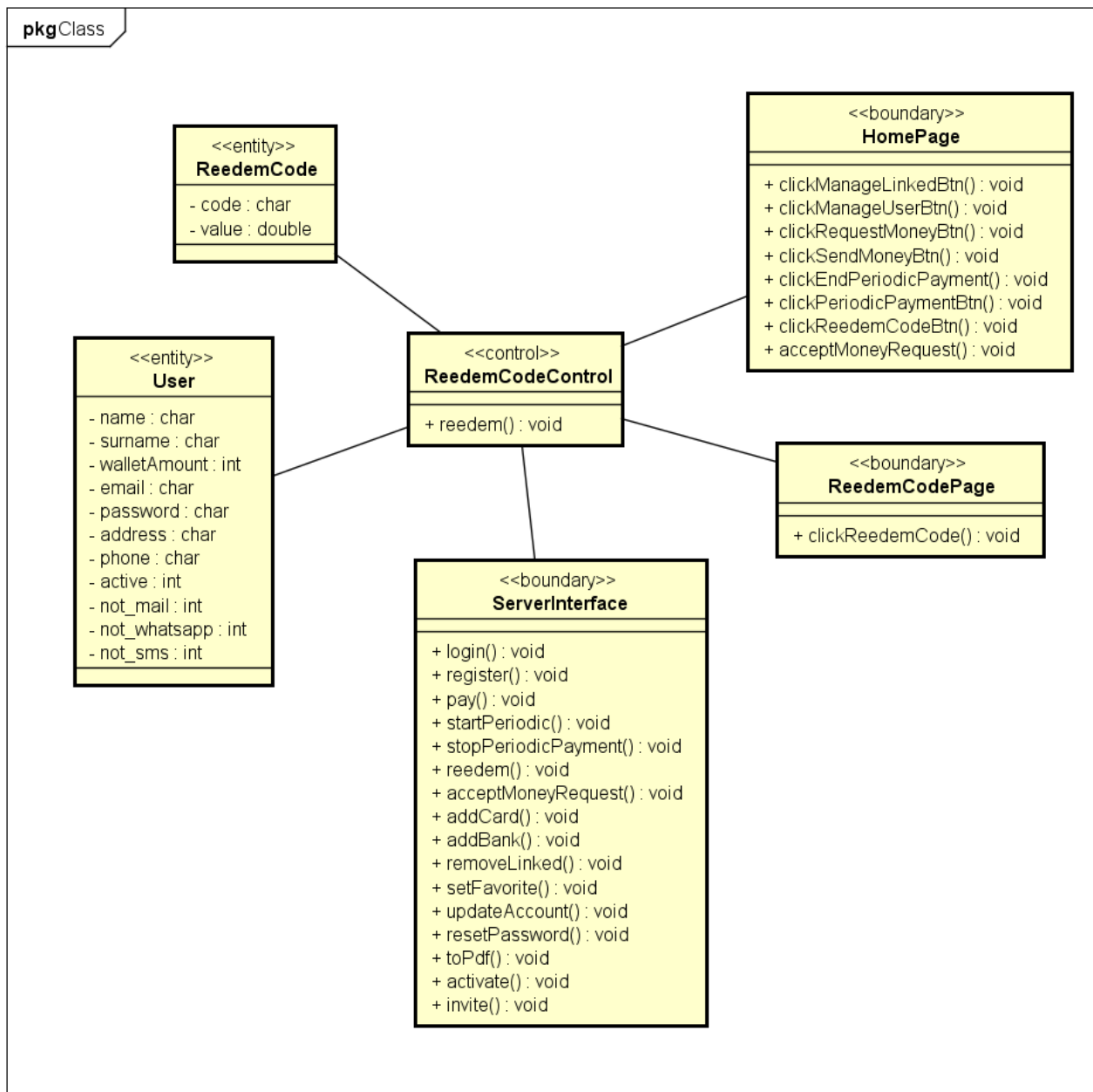
Nome caso d'uso:	Reedem_code
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "Riscatta codice" 2. Il sistema mostra la schermata per il riscatto dei codici e mostra i codici creati dall'utente e non ancora riscattati 3. L'utente inserisce il codice e fa click su "Riscatta" 4. La somma corrispondente al codice viene aggiunta all'account ed il server invia una notifica al creatore del codice per informarlo che quel codice non è più disponibile
Post condizioni:	Viene mostrato un messaggio di conferma e viene ricaricata la pagina iniziale

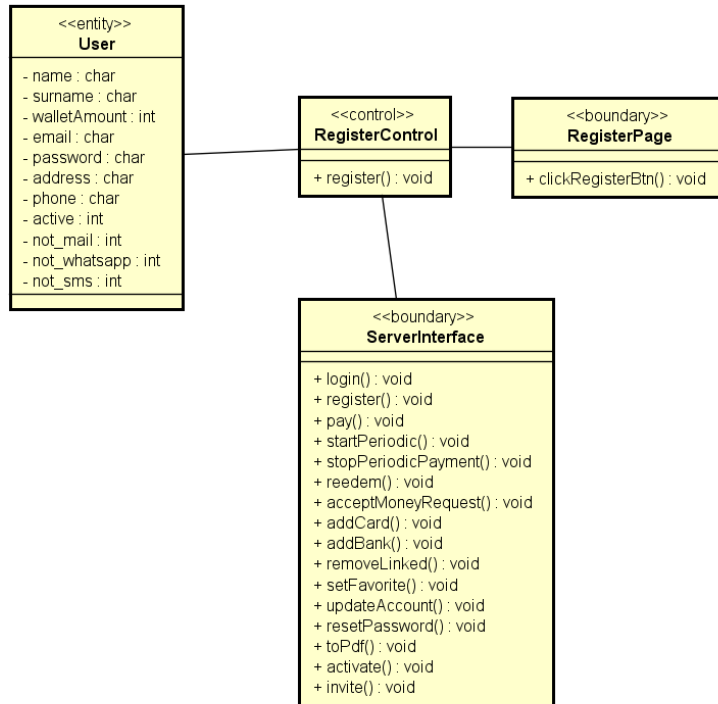
Nome caso d'uso:	Accept_money_request
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	<ol style="list-style-type: none"> 1. L'utente fa click su "Notifiche" 2. Il sistema mostra le notifiche ricevute 3. L'utente sceglie la richiesta desiderata e fa click su "Paga"

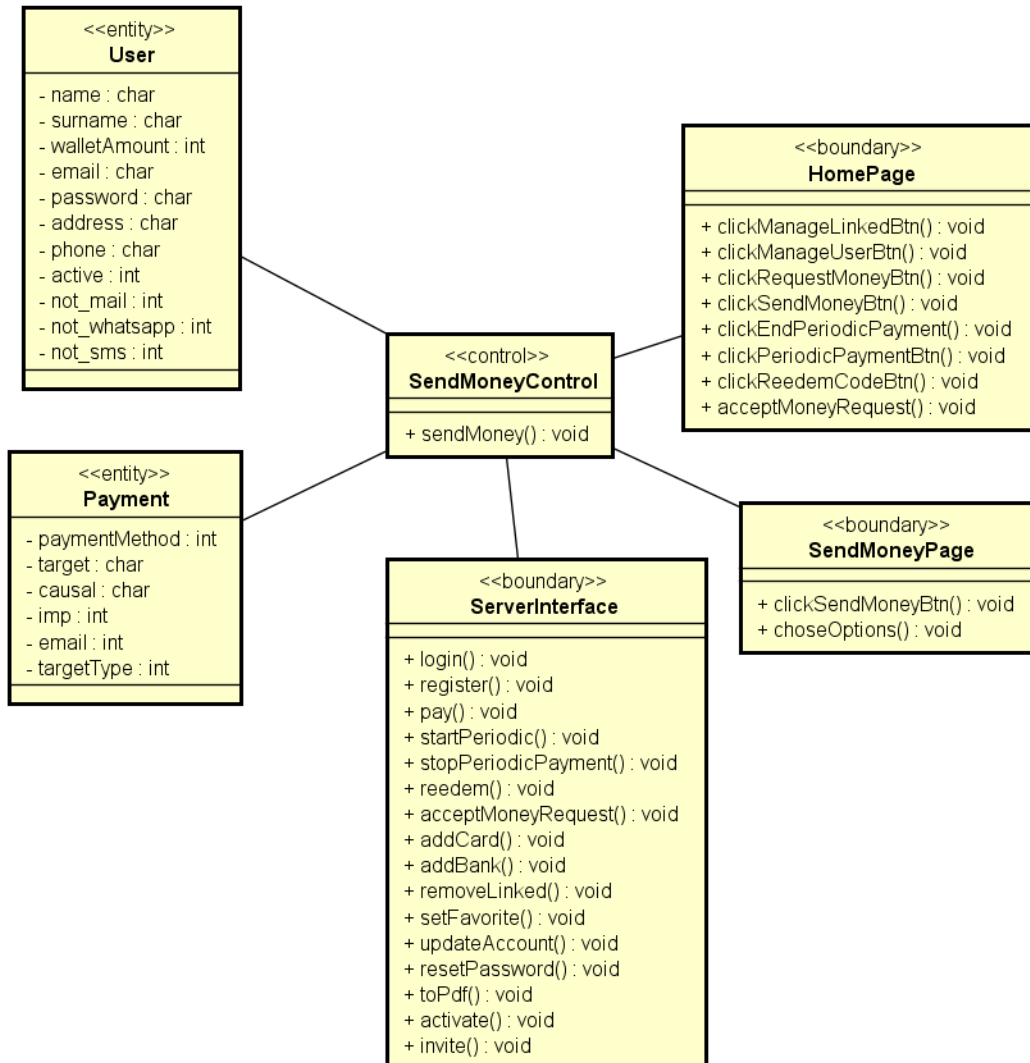
	4. Il sistema comunica al server di inviare i soldi all'utente richiedente, inviando una notifica al richiedente dell'avvenuto pagamento
Post condizioni:	Viene mostrato un messaggio di conferma e viene ricaricata la pagina iniziale

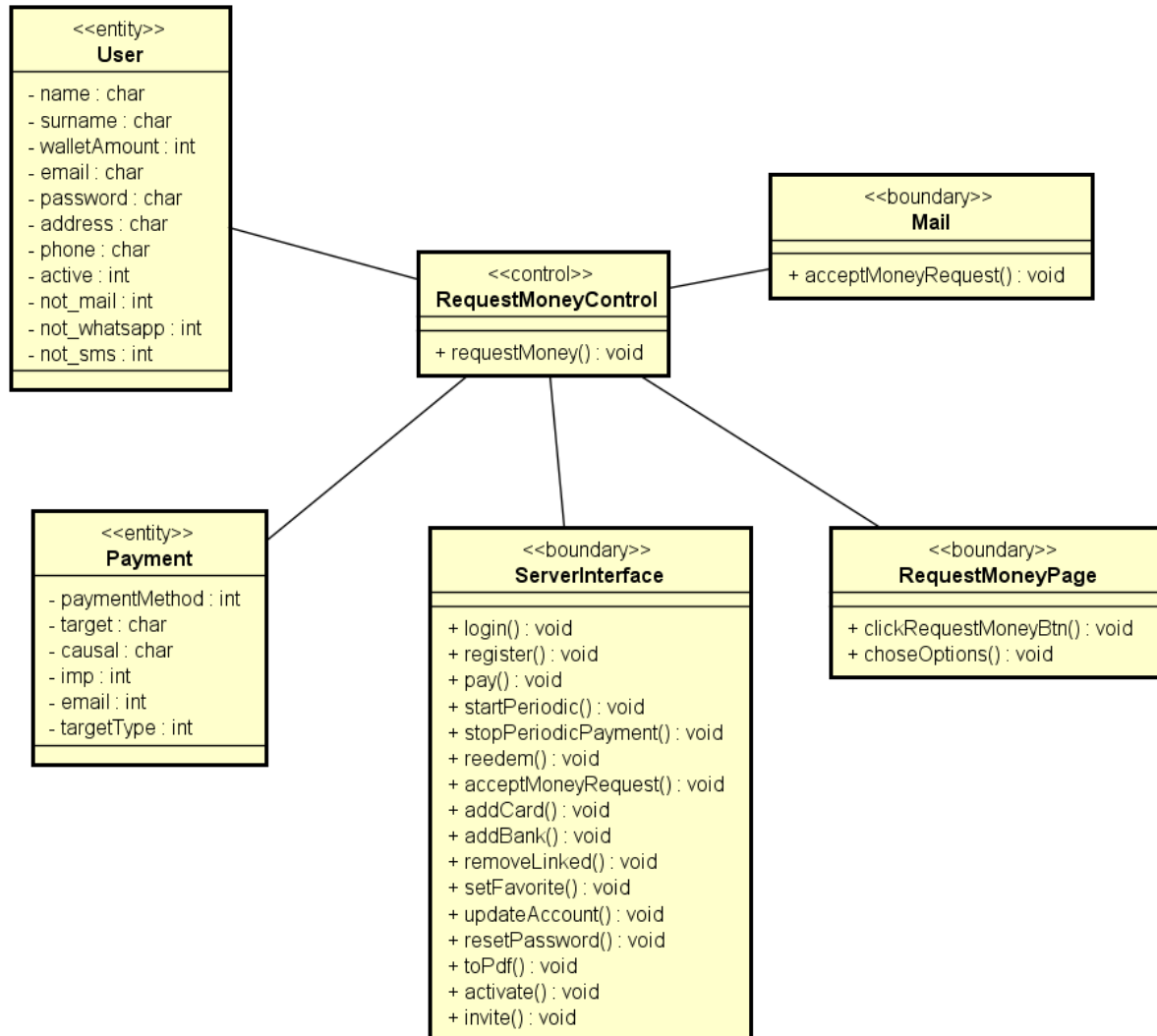
Nome caso d'uso:	Print_PDF
Attori:	Utente, server
Pre Condizioni:	Il sistema mostra la schermata home
Flusso degli eventi:	1. L'utente fa click su "Esporta movimenti in PDF"
Post condizioni:	Viene scaricato il file PDF con i report

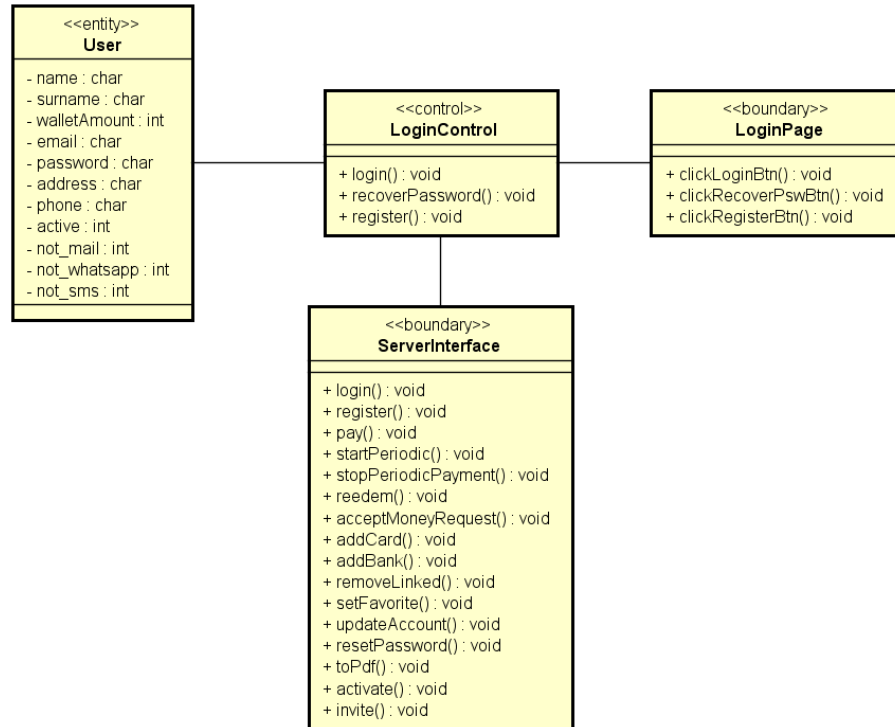
3.3 Diagrammi delle classi

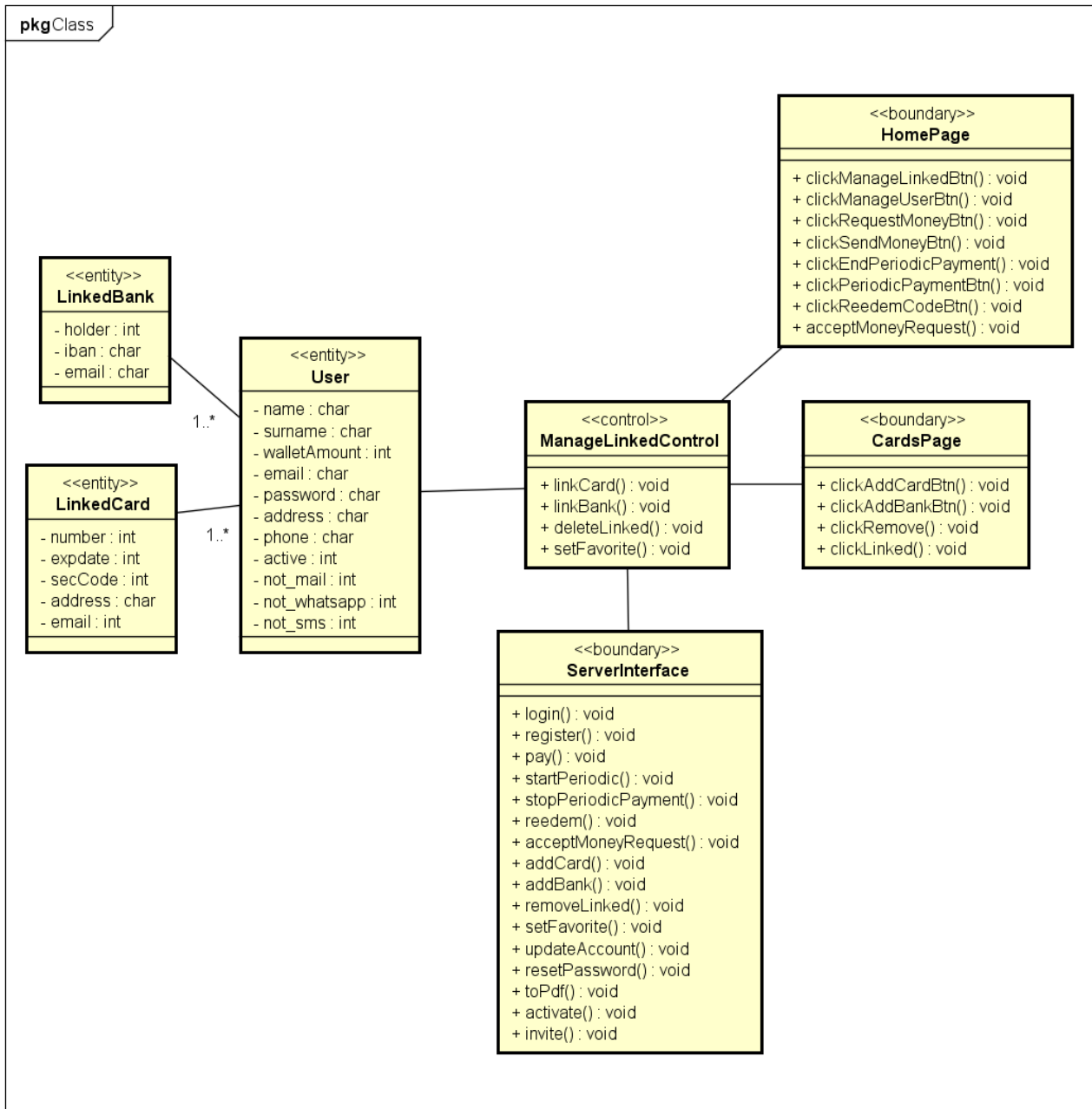


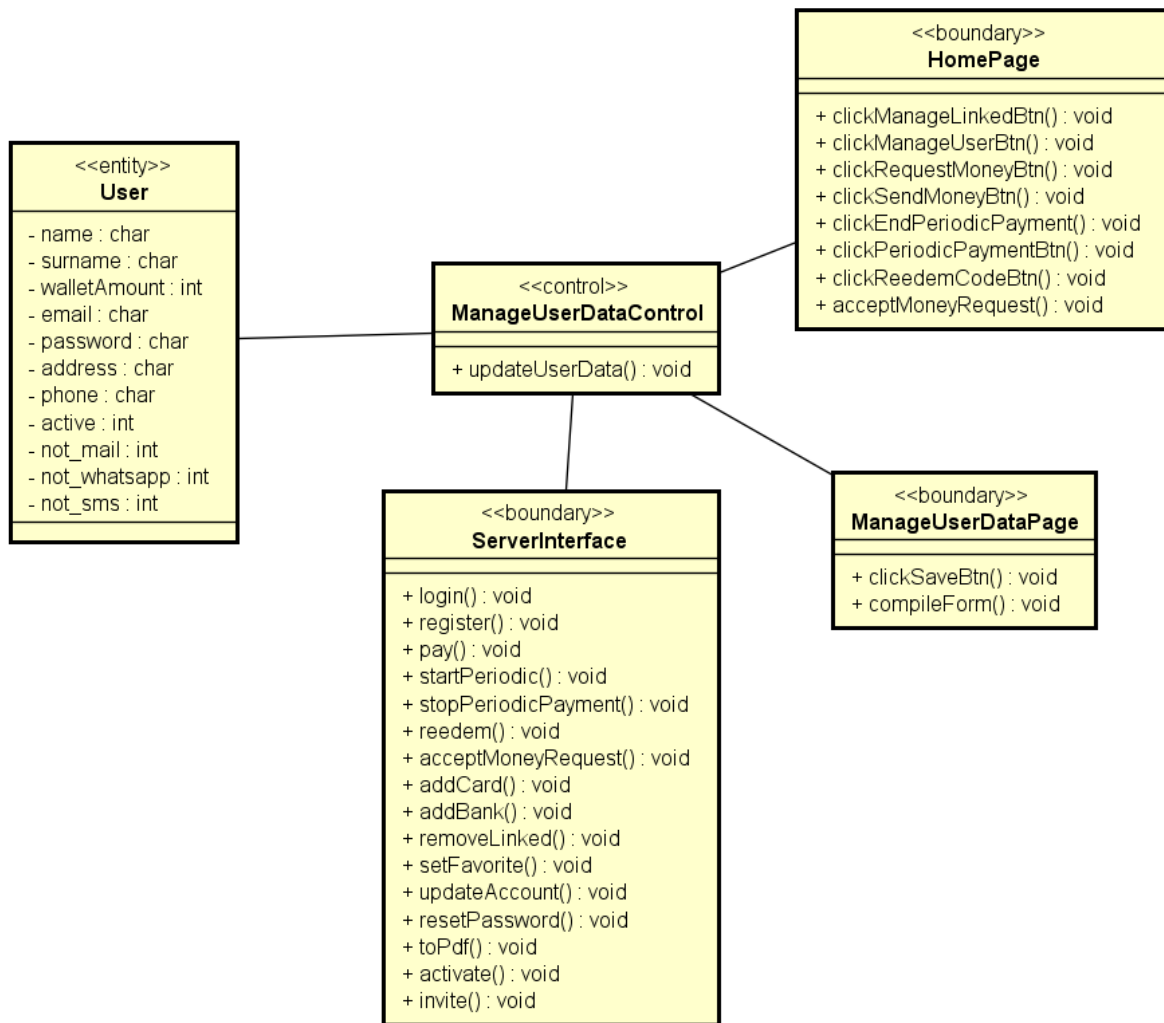


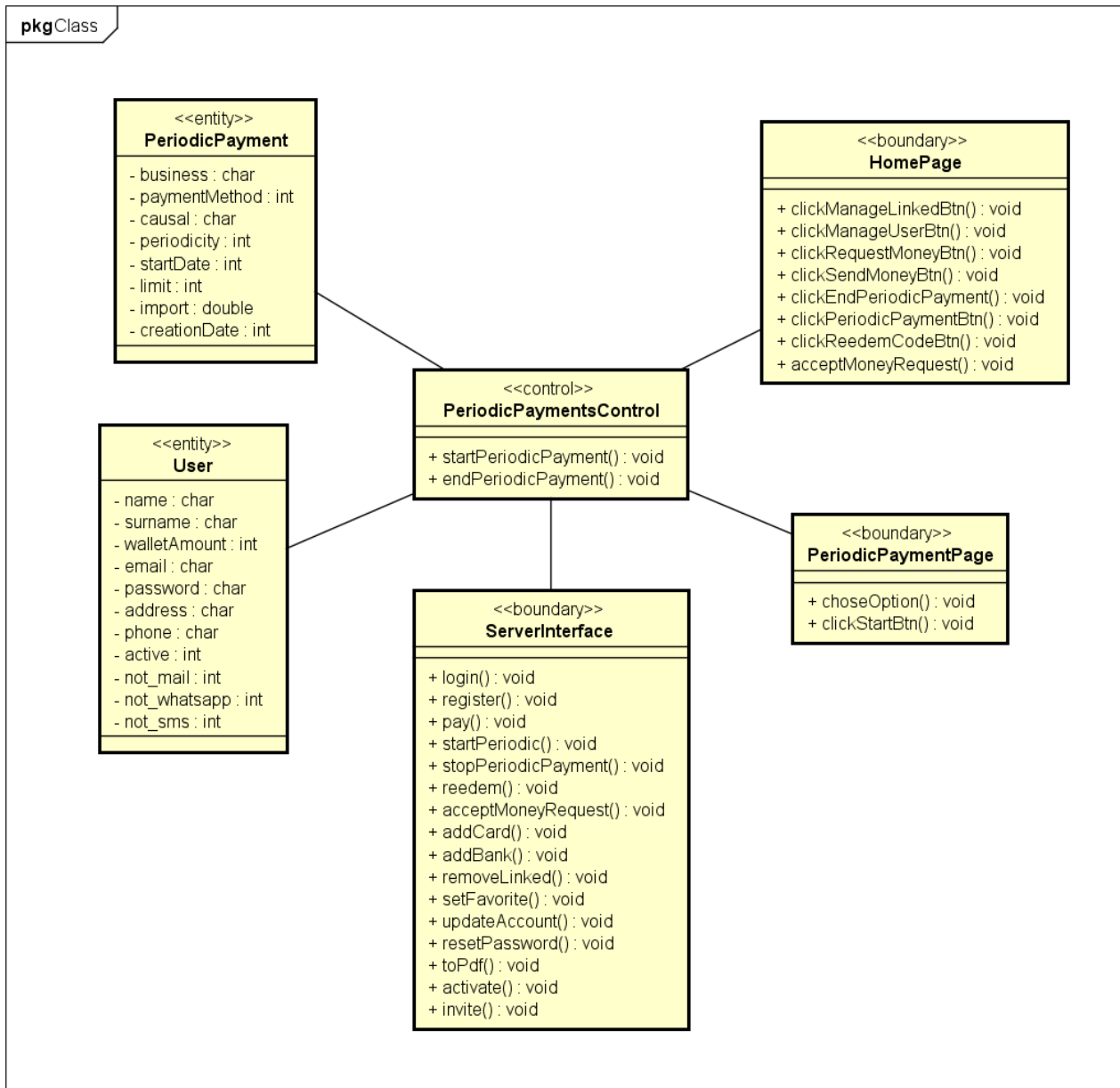




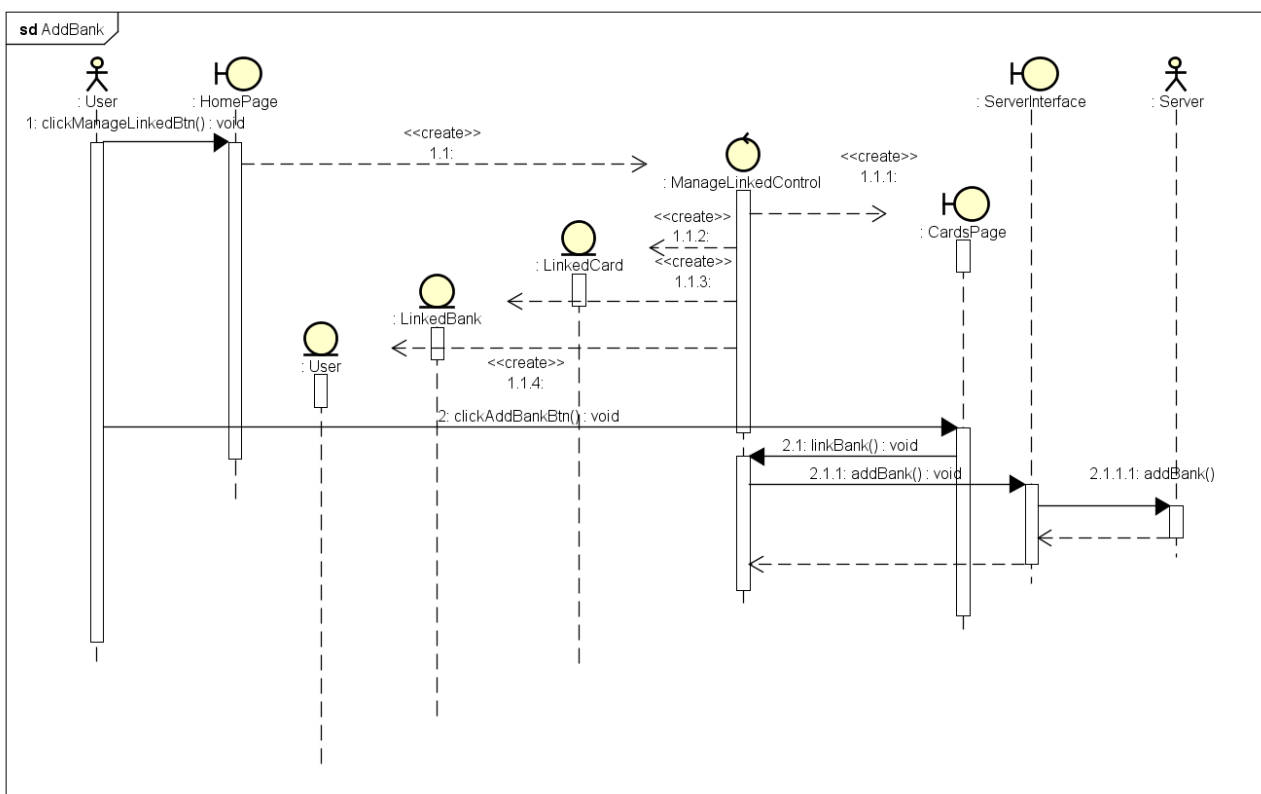
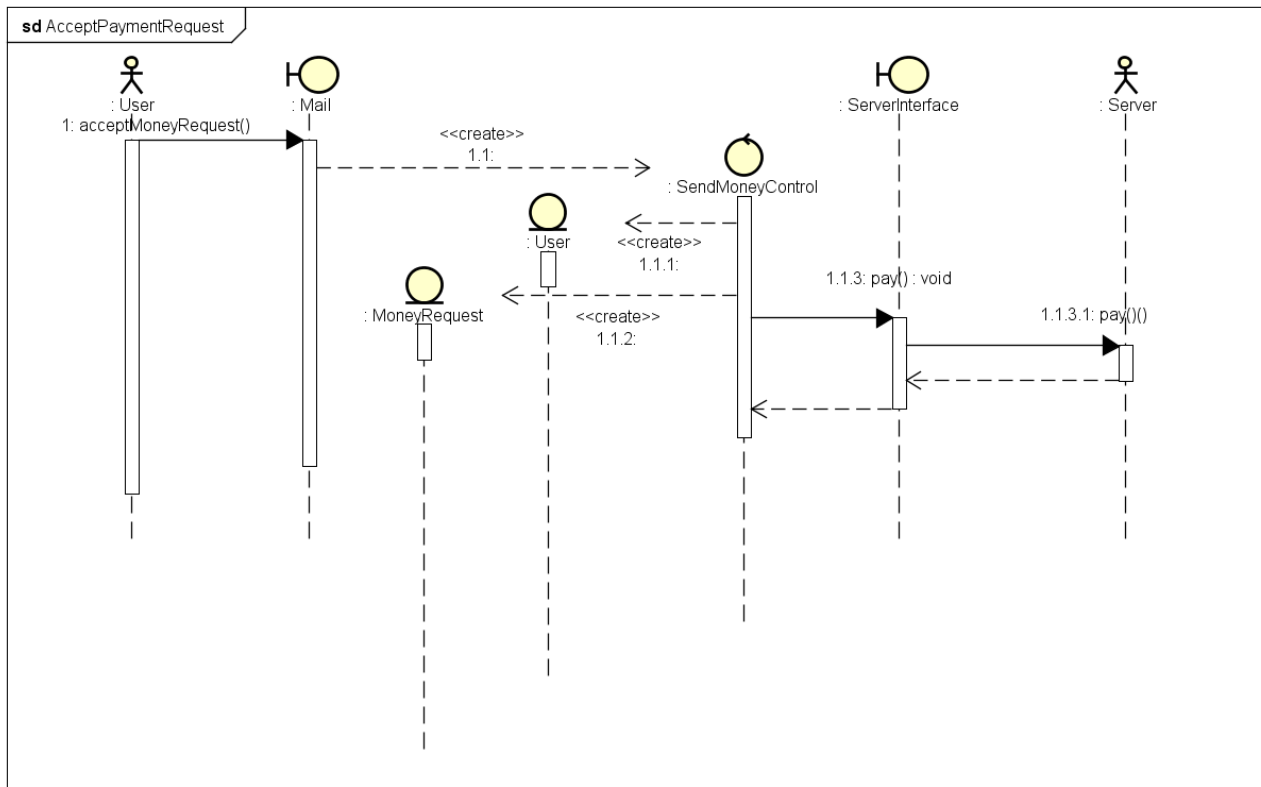


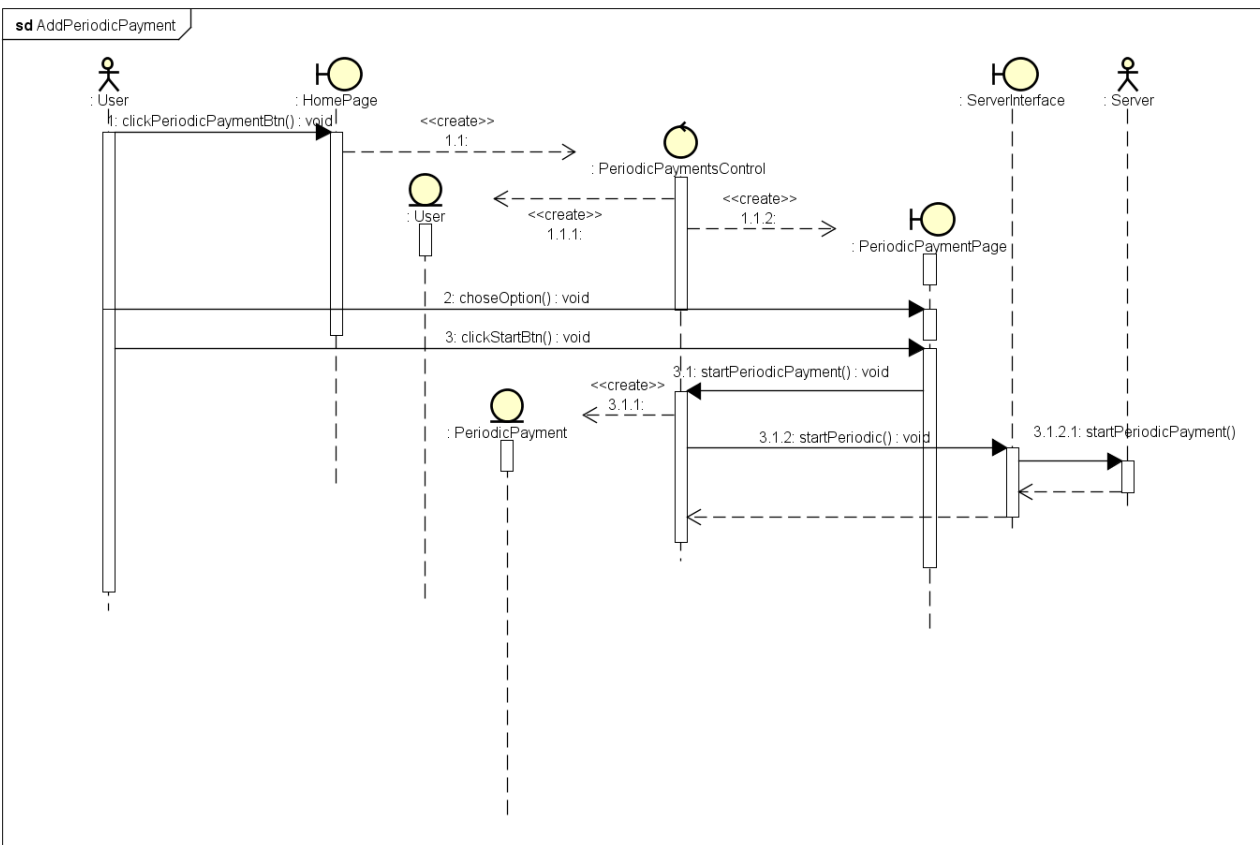
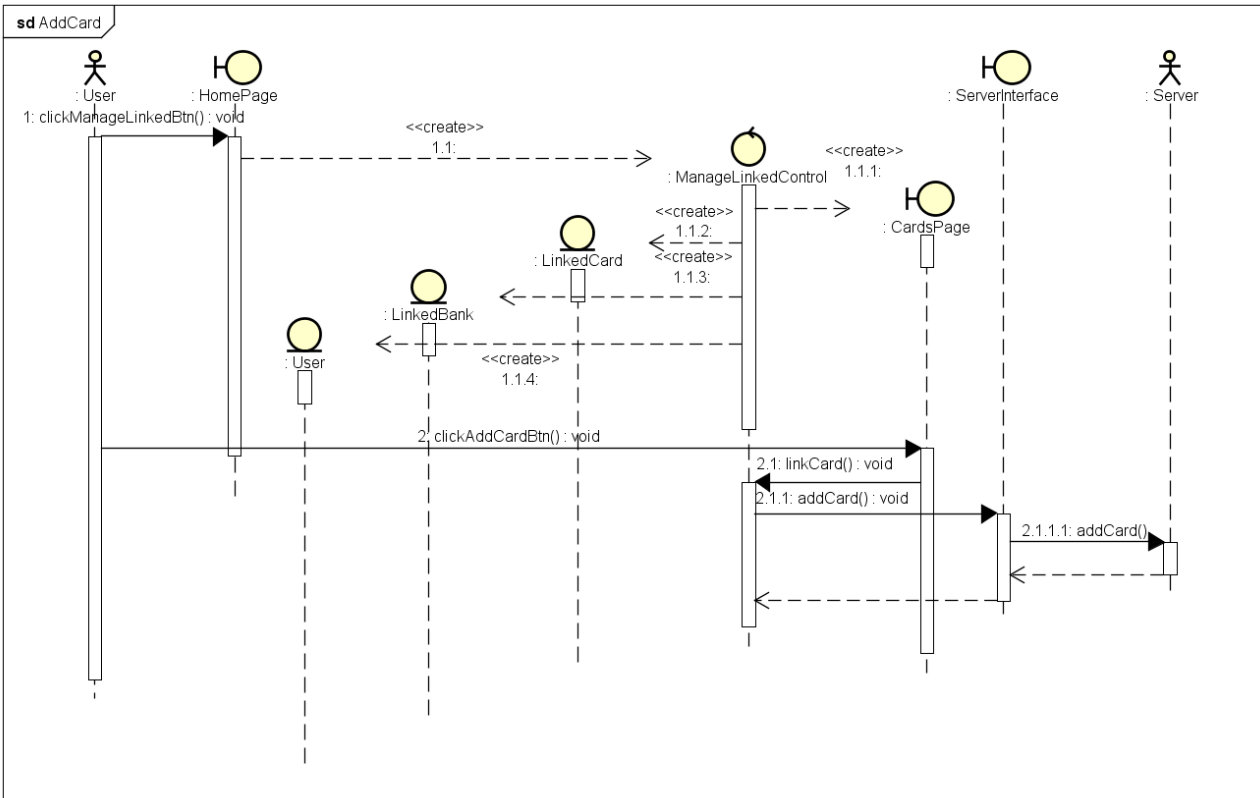


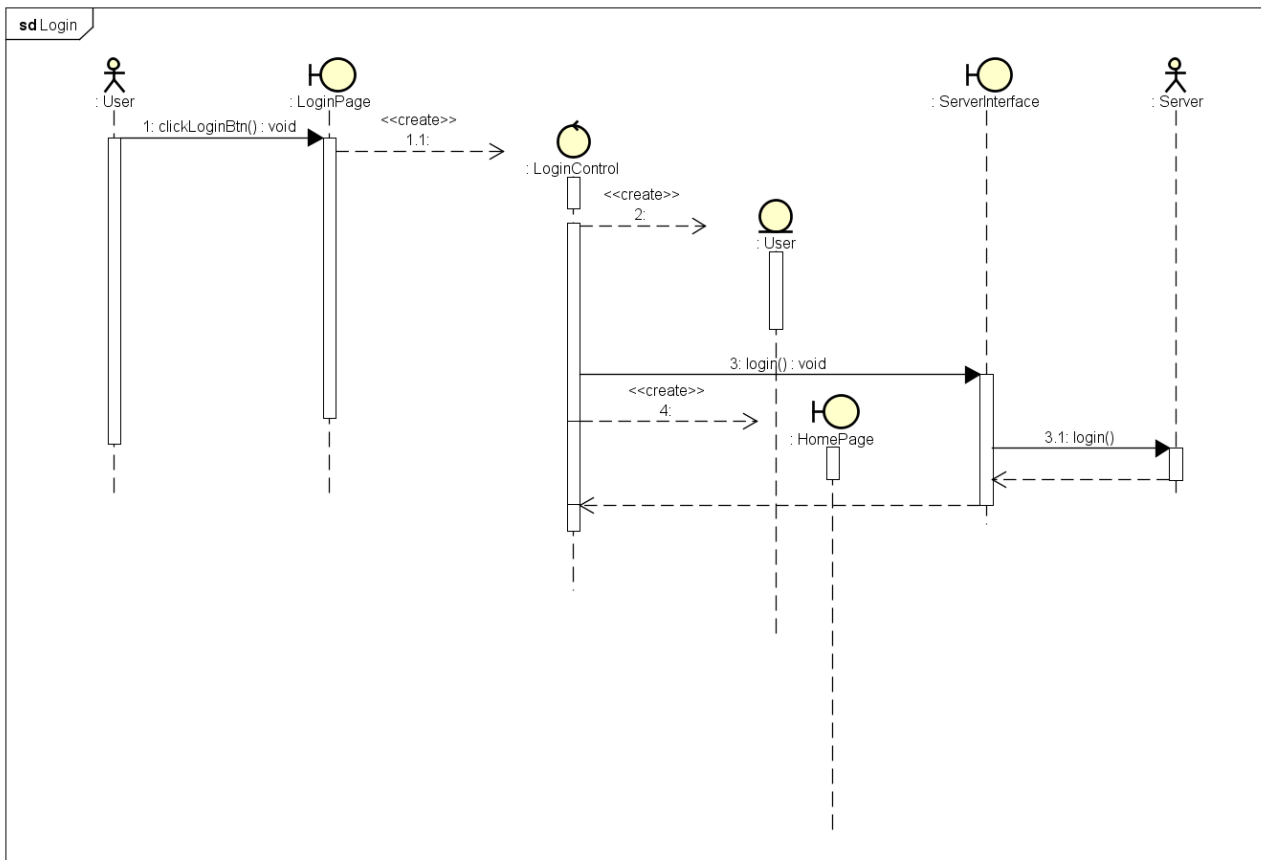
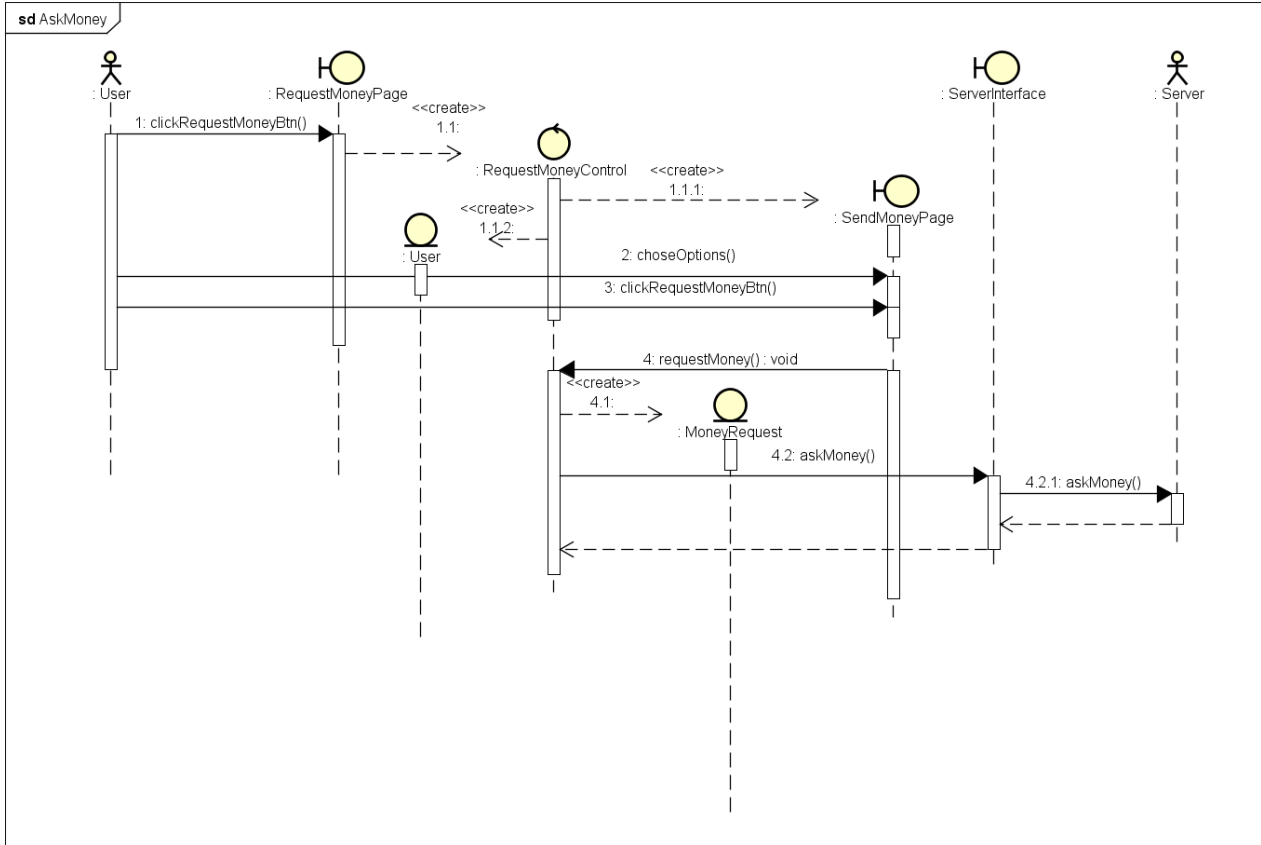


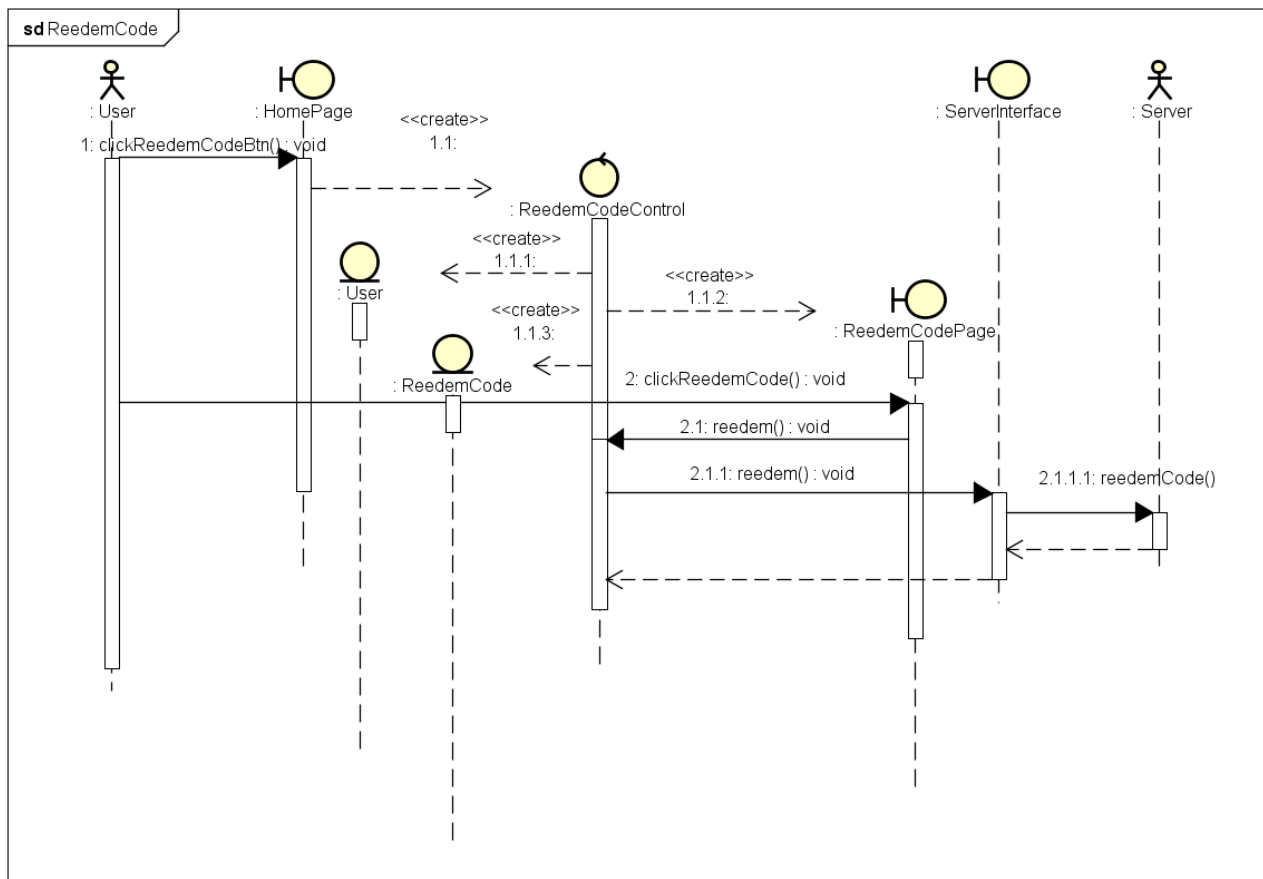
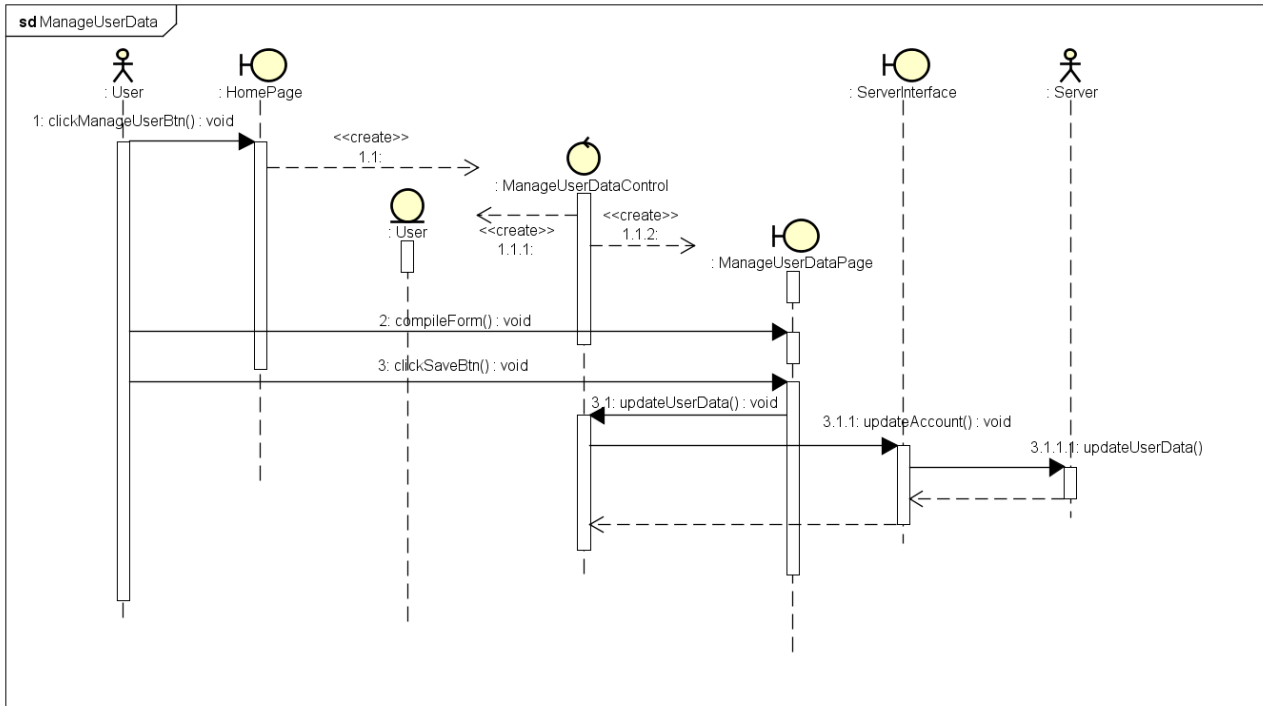


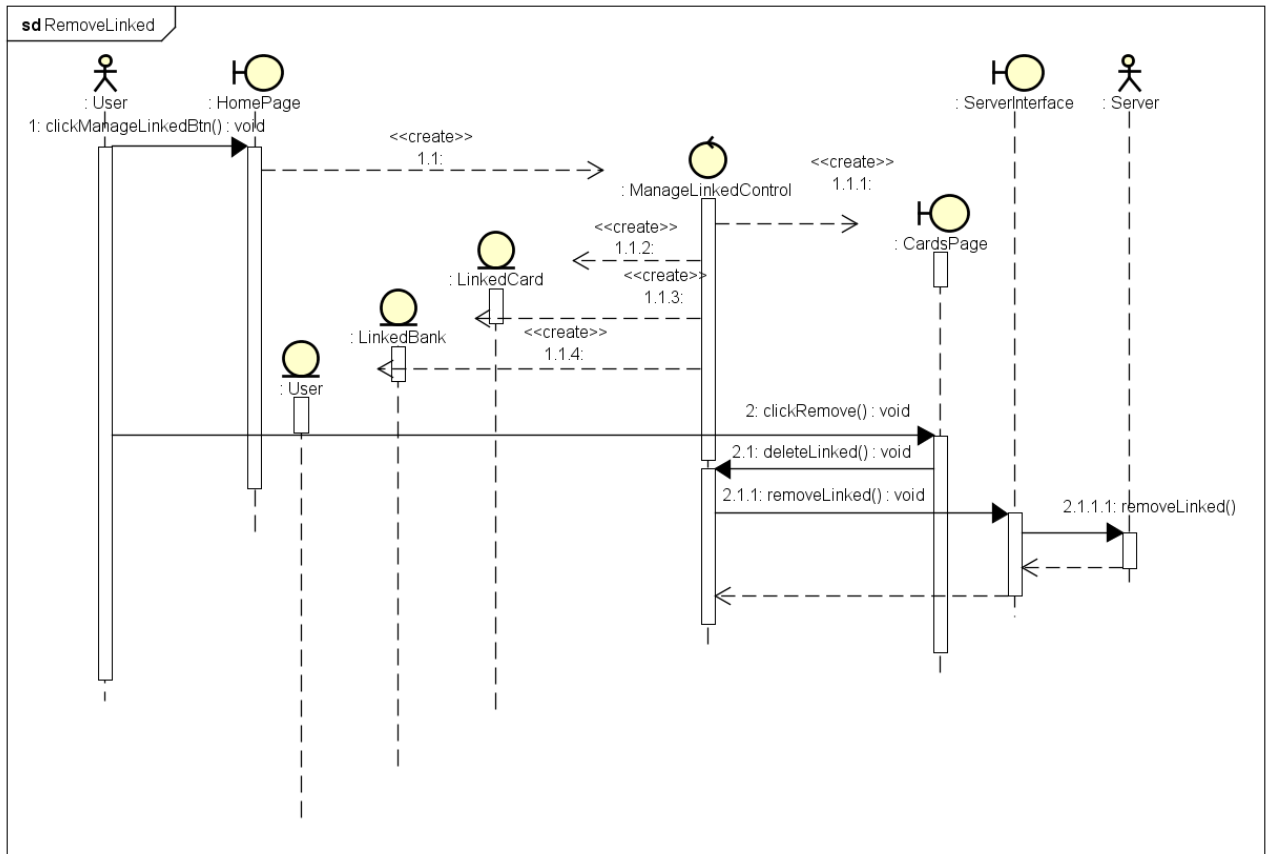
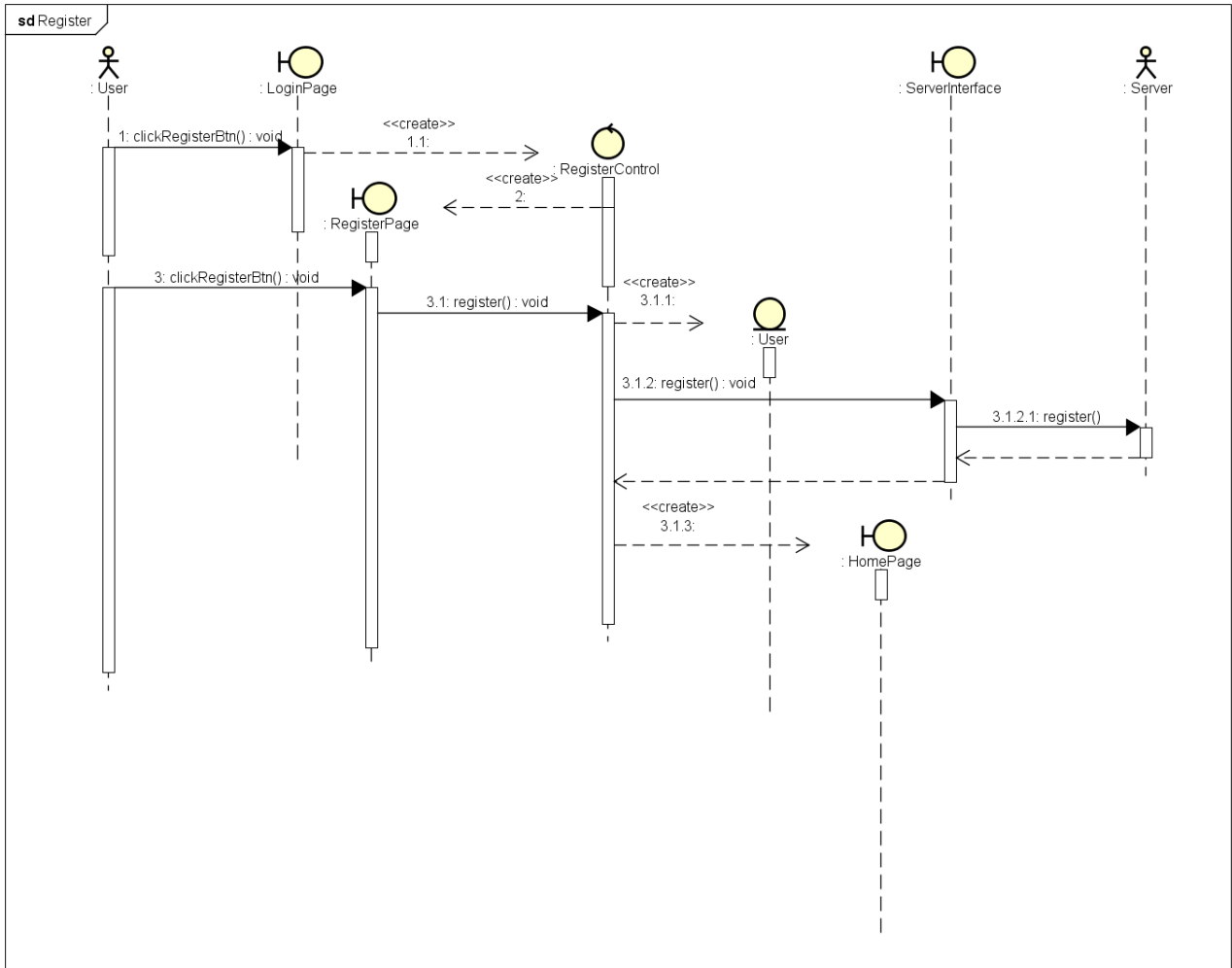
3.4 Diagrammi di sequenza



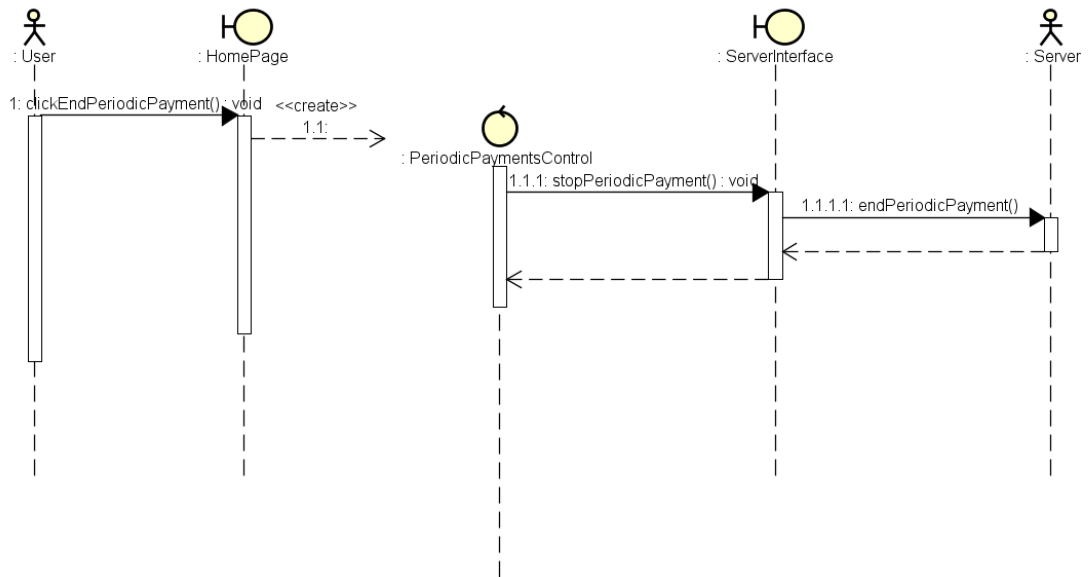




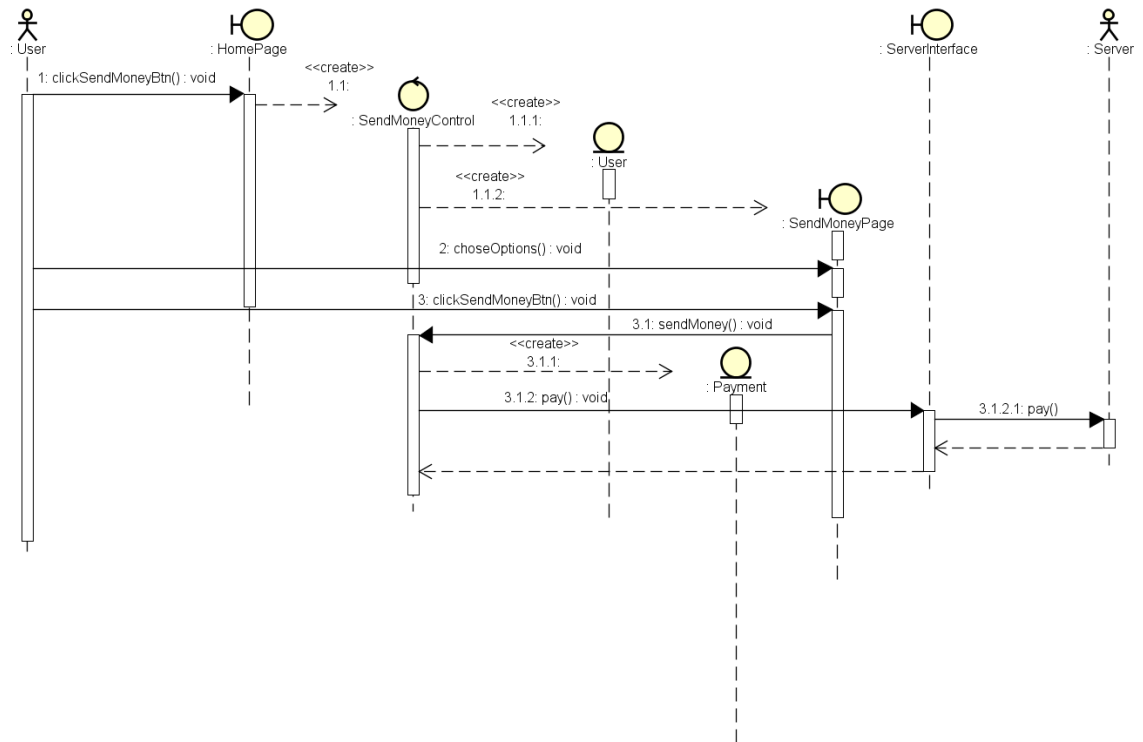


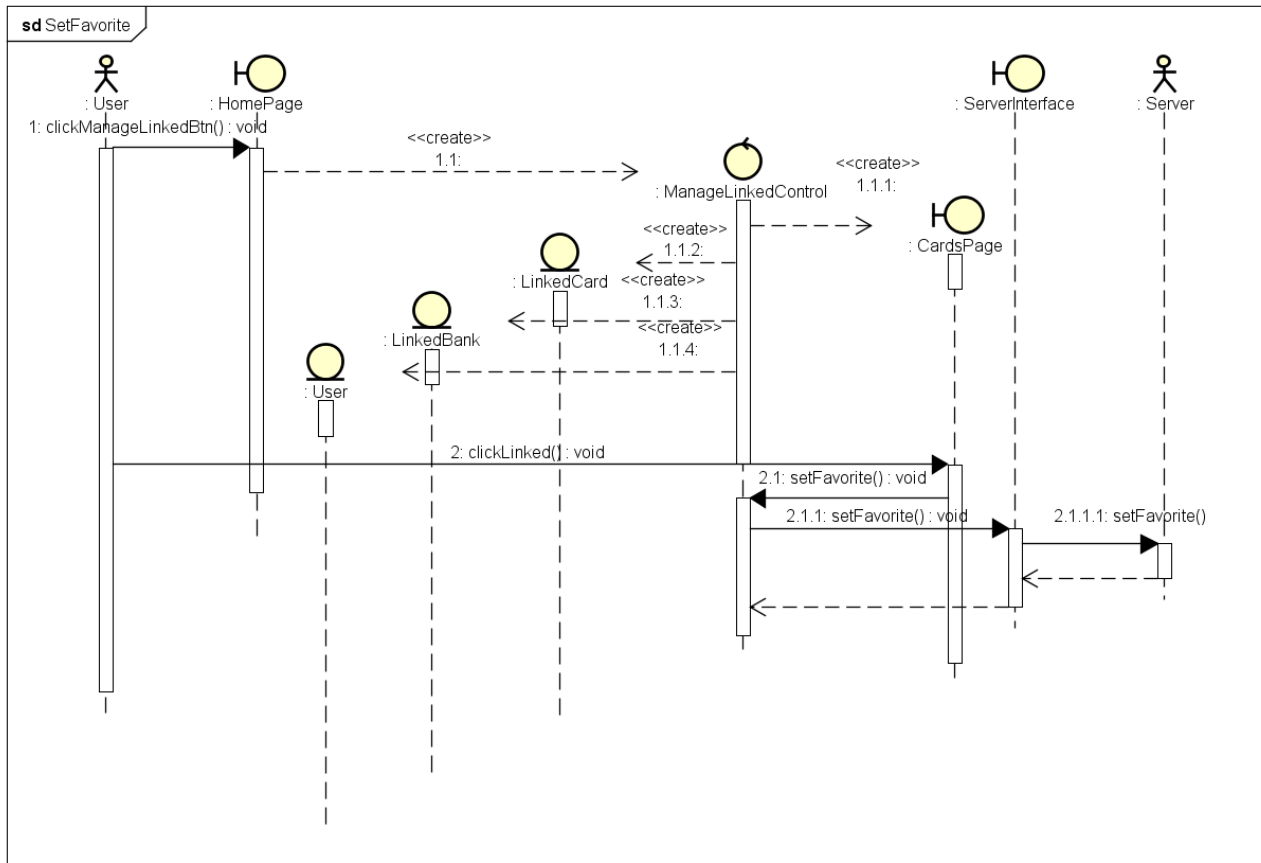


sd RemovePeriodicPayment



sd SendMoney





4. Object Design

4.1 Trade off

- Il sistema utilizza lato client il protocollo HTTP per richiedere pagine HTML associate a fogli di stile CSS e script Javascript con l'ausilio di JQuery. Questa soluzione permette in maniera semplice di accedere ai servizi tramite qualsiasi browser, inoltre questa scelta rende compatibile l'interfaccia per un'implementazione mobile e relativo sviluppo di un'applicazione android in formato APK
- Lato server viene utilizzato il linguaggio Javascript tramite Node.js. In particolare viene utilizzato il modulo Express per la creazione del web server e la gestione delle richieste. Il modulo MySQL per la comunicazione col database, il modulo Email per l'invio delle notifiche tramite email ed il modulo Twilio per l'invio di messaggi Whatsapp ed SMS.

4.2 Descrizione dei model

- **Entity:**
 - *LinkedBank:*
Contiene i dati sull'account bancario collegato
 - *LinkedCard:*
Contiene i dati sulle carte collegate
 - *MoneyRequest:*
Contiene i dati necessari ad effettuare richieste di pagamento
 - *Payment:*
Contiene i dati necessari ad effettuare pagamenti
 - *PeriodicPayment:*
Contiene i dati necessari ad effettuare pagamenti periodici
 - *RedemCode:*
Contiene i dati per riscattare codici
 - *User:*
Contiene tutti i dati riguardanti l'utente

4.3 Descrizione dei view

- **Boundary:**
 - *CardsPage:*
Rappresenta la pagina dove è possibile aggiungere o rimuovere carte o account bancari
 - *HomePage:*
Rappresenta la pagina principale dove è possibile accedere alle varie funzioni
 - *LoginPage:*

Rappresenta la pagina dove è possibile effettuare l'accesso

- *ManageUserDataPage:*

Rappresenta la pagina dove è possibile modificare i dati dell'utente

- *Mail:*

Rappresenta la pagina dal quale si può accettare una richiesta di pagamento

- *PeriodicPaymentPage:*

Rappresenta la pagina dove è possibile avviare un pagamento periodico

- *ReedemCodePage:*

Rappresenta la pagina dove è possibile visualizzare e riscattare i codici

- *RegisterPage:*

Rappresenta la pagina dove è possibile registrarsi

- *RequestMoneyPage:*

Rappresenta la pagina dove è possibile inviare richieste di pagamento

- *SendMoneyPage:*

Rappresenta la pagina dove è possibile inviare richieste di pagamento

- *ServerInterface:*

Interfaccia di comunicazione con le funzionalità offerte dal server

4.4 Descrizione dei control

- **Control:**

- *LoginControl:*

Si occupa della gestione del login dell'utenza

- *ManageLinkedControl:*

Si occupa della gestione delle carte e gli account bancari collegati

- *ManageUserDataControl:*

Si occupa della gestione dei dati dell'utente e della gestione delle notifiche

- *PeriodicPaymentsControl:*

Si occupa della gestione dei pagamenti periodici

- *ReedemCodeControl:*

Si occupa della gestione dell'area riscatto codici

- *RegisterControl:*

Si occupa della gestione delle registrazioni dei nuovi utenti

- *RequestMoneyControl:*

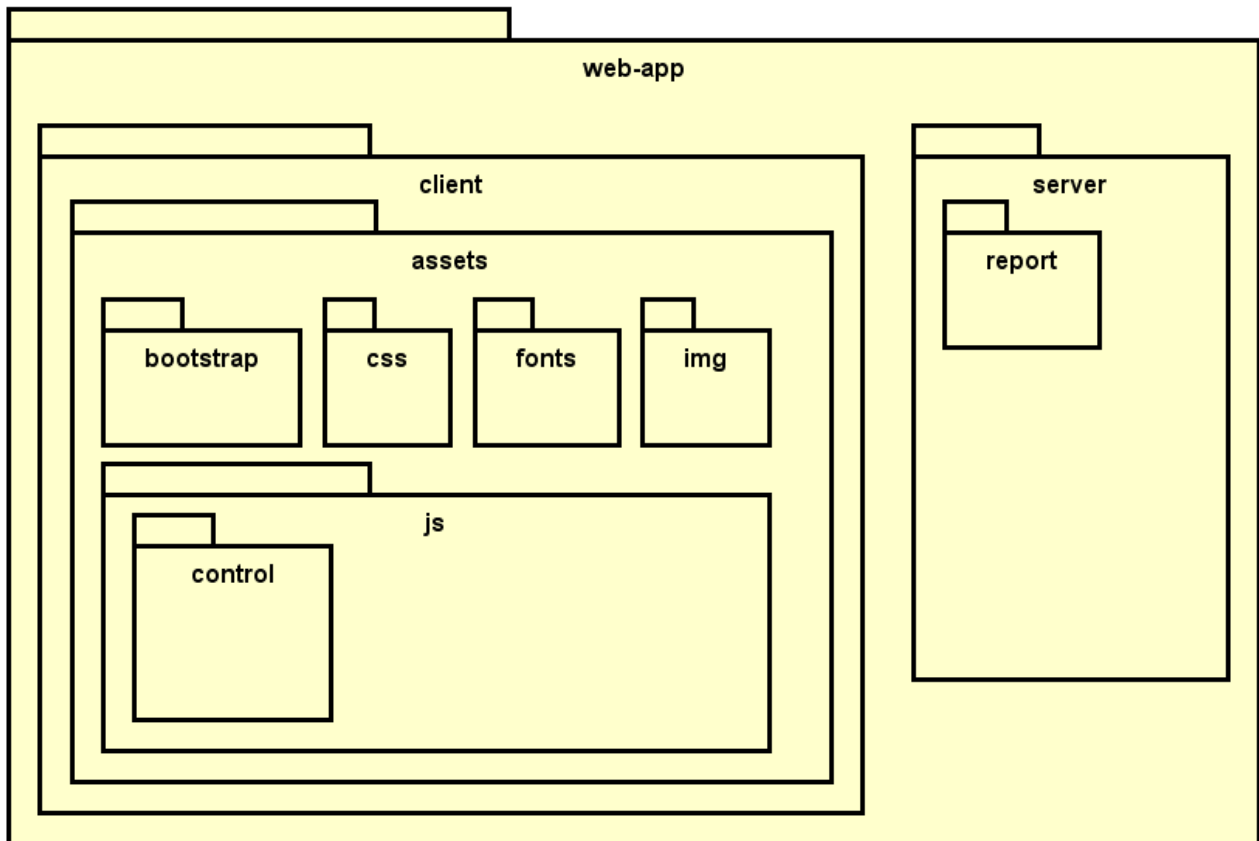
Si occupa della gestione delle richieste di denaro

- *SendMoneyControl:*

Si occupa della gestione di invio di denaro

5. Package

5.1 Organizzazione dei package



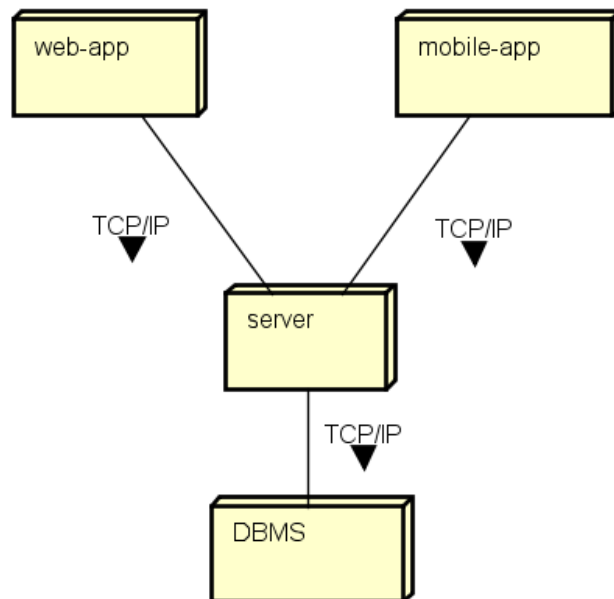
Descrizione dei package:

- **web-app:** Package generale
- **Client:** Package che racchiude tutte le risorse lato client, ed i file html
- **Server:** Package che racchiude tutte le risorse lato server, ovvero il file sorgente index.js
- **Assets:** Contiene le risorse per i file html contenute in client
- **Bootstrap:** Contiene le librerie del pacchetto grafico bootstrap
- **CSS:** Contiene le risorse CSS
- **Fonts:** Contiene i font aggiunti
- **Img:** Contiene le immagini in formato jpg e png
- **Js:** Contiene i file script Javascript generici
- **Control:** Contiene i file script Javascript con la definizione delle funzioni delle control

6. System Design

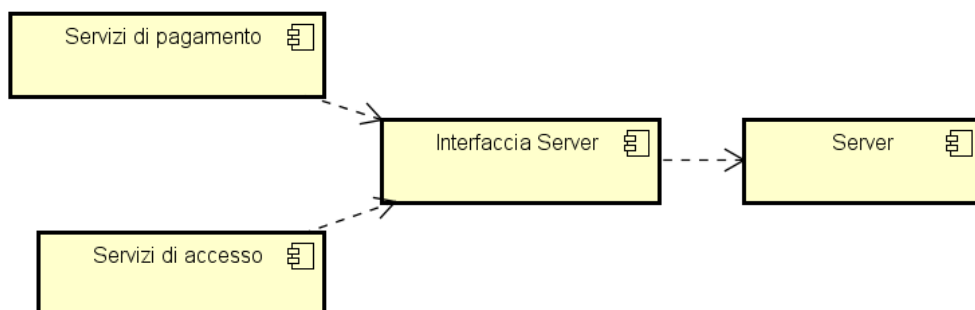
6.1 Hardware – Software mapping

Il sistema è composto da quattro componenti principali che comunicano tra di loro tramite connessioni internet TCP/IP tramite porta 8001. In particolare l'applicazione web e l'applicazione mobile comunicano col server tramite richieste asincrone di tipo GET e POST. Il server le elabora e risponde alle richieste, e se necessario instaura connessioni col database tramite TCP/IP sulla porta 3306.



6.2 Decomposizione del sistema

Data l'architettura del sistema, le componenti interagiscono tra loro al fine di adempiere alle funzionalità prefissate. Di conseguenza si consegue dall'immagine sottostante che ogni componente dialoga con L'interfaccia server al fine di recuperare le informazioni necessarie per gli utenti.



7. Database Design

7.1 Diagramma ER ristrutturato

